AMENDMENTS TO SPECIFICATION

Please replace the original title with the following amended title:

--PROCESSOR AND METHOD CAPABLE OF EXECUTING CONDITIONAL INSTRUCTIONS BASED ON A FLAG --

Please replace the indicated paragraphs with the following amended paragraphs:

Page 1, lines 5-7:

The present invention relates to the technical field of processors and, more particularly, to a processor capable of executing conditional instructions based on a flag.

Page 1, line 16 to Page 2, line 10:

To eliminate the above problem, U.S. Patent 5,961,633 granted to Jaggar for ~~an~~ "Execution of ~~data processing instructions~~ Data Processing Instructions" uses a 4-bit (from 31-st to 28-th bits) condition field and 28-bit (from 27-th to 0-th bits) operation field ~~on~~ in instruction decode. Also, a condition tester is applied to test the condition field and four flags N, Z, C, V of the processor, to produce an output signal to determine whether to discard the instruction or not. The operating process is illustrated in FIG. 1, which shows C programming codes. FIG. 2 is a schematic view of instructions in machine codes after the C programming codes of FIG. 1 are compiled and assembled. When the processor executes the instruction (1), the Z flag of the processor is set, as the content of register R1 is 0. When the processor executes the instruction (2), the condition field of the instruction (2)

is ~~EQ~~ "EQ". The condition tester tests the condition field and thus obtains a state that is the same as the Z flag. Therefore, no output signal is produced and the instruction (2) is normally executed by the processor. When the processor executes the instruction (6), the condition field of the instruction (6) is ~~NE~~ "NE". The condition tester tests the condition field and thus obtains a state that is different from the Z flag, so that the output signal is produced, and although the instruction (6) is executed by the processor, the result is discarded.

Page 2, lines 11-14:

When the processor executes the C programming codes shown in FIG. 1, the instructions (1) to (10) are performed. When the content of R1 is 0, the results performed on the instructions (6) to (9) are discarded, and otherwise, the results performed on the instructions (2) to (5) are discarded.

Page 2, line 21 to Page 3, line 5:

However, such a processor requires 4-bit condition field in an instruction. For a 16-bit instruction, only 12 bits are remained in use for encoding. This does not meet with the typical instruction number requirement. Therefore, such a design of condition ~~field~~ fields is not existed in a 16-bit instruction. Moreover, in this conventional skill, no matter what the result of the conditional instruction, the subsequent instructions have to be performed but some of the results are discarded. This also adds the load of the processor. Therefore, the design of conditional instruction processing for the processor in the prior art is not satisfactory.

Page 3, lines 7-12:

The object of the present invention is to provide a processor and method capable of executing conditional instructions based on a flag, which ~~increase~~ increases the efficiency of a processor with a pipeline processing in using a branch or jump instruction, and which also prevents using a long encoding field and avoids occupying the pipeline processing time when no instruction is performed, so as to increase the code density and performance.

Page 3, line 13 to Page 4, line 6:

According to a feature of the present invention, there is provided a processor capable of executing conditional instructions based on a flag. The instruction set executed by the processor includes M-bit instructions and N-bit instructions (where M,N are positive integers, M>N). the instruction set has condition execution instructions and M-bit parallel condiction execution instructions. The parallel condition execution instruction has a first N-bit instruction and a second N-bit instruction. The processor comprises: a flag having a state; an instruction fetching device, to fetch at least one instruction to be performed; an instruction decoder, to decode the instruction fetched by the instruction fetching device; an instruction executing device, to execute the instruction outputted by the instruction decode, wherein the state of the flag is set according to a result of executing a condition execution instruction, which indictates a state of condition acceptance or rejection; and a mode switching device, to switch the instruction decoder to decode one of the first and the second N-bit instructions according to the state of the flag, so as to be subsequently performed

by the instruction executing device, when a prallel condition execution instruction is fetched by the instruction fetching device.

Page 4, lines 7-21:

According to another feature of the present invention, there is provided a method capable of executing conditional instructions in a processor based on a flag. The instruction set executed by the processor includes M-bit instructions and N-bit instructions (where M,N are positive integers, M>N). the instruction set having condition execution instructions and M-bit parallel condition execution instructions. The parallel condition execution instruction has a first and a second N-bit ~~instructions~~ instruction. The method comprises: (A) fetching at least one instruction to be decoded and executed; (B) when a condition execution instruction is performed, setting a flag to a first logic state if the execution results in a condition acceptance, and setting the flag to a second logic state if the execution results in a condition rejection; and (C) when the instruction fetched is a parallel condition execution instruction, decoding and executing the first N-bit instruction if the flag is ~~on~~ in the first logic state, and decoding and execting the second N-bit instruction if the flag is ~~on~~ in the second logic state.

Page 5, line 18 to Page 6, line 10:

FIG. 3 is a block diagram of a processor capable of executing conditional instructions based on a flag in accordance with the invention. The processor includes a flag 310, an instruction fetching device 320, an instruction decoder 330, an instruction executing device

340 and a mode switching device 350. The instruction fetching device 320 is provided to fetch at least one instruction to be performed. The instruction set performed by the processor includes M-bit instructions and N-bit instructions (where M, N are positive integers, M>N, e.g., M=32 and N=16). In addition to the general M-bit and N-bit instructions, the instruction set also has N-bit or M-bit condition execution instructions (for example, compare instruction) and M-bit parallel condition execution instructions. Each parallel condition execution instruction is an M-bit instruction with at least two N-bit instructions. As shown in FIG. 4, a 32-bit parallel condition execution instruction has a first N-bit (N=16) instruction and a second N-bit instruction, wherein the result of executing the condition execution instruction determines ~~which of~~ whether to execute the first or second N-bit instruction ~~to be executed~~.

Page 6, lines 11-18:

The instruction decoder 330 decodes the instruction fetched by the instruction fetching device 320. The instruction executing device 340 executes the instruction outputted by the instruction decoder 330. When the executed instruction is an N-bit or M-bit condition execution instruction, the instruction executing device 340 sets the flag 310 according to the result of executing the condition execution instruction. Namely, the flag 310 is set to "true" when the result performed on the condition execution instruction ~~is~~ indicates a state of condition acceptance, and conversely to "false".

Page 7, lines 6-22:

FIG. 5 shows an embodied example. In FIG. 5, C programming codes of FIG. 1 are compiled and assembled into a schematic view of instructions in the form of machine codes. As shown in FIG. 5, the instruction (1) is an M-bit (M=32) condition execution instruction (compare instruction). When the processor executes the instruction (1) and the content of register R1 is [[0]] zero, the comparison obtains the same result, so that the result performed on the condition execution instruction is the state of condition acceptance. Therefore, the flag 310 is set to "true". At this point, when the processor executes the parallel condition execution instruction (2), the processor finds the flag as ~~true~~ "true" and thus executes only the first N-bit instruction [MOVEQ R1, R5] without executing the second N-bit instruction [MOVNE R1, R9]. Similarly, for subsequent parallel condition execution instructions (3)~(5), the processor executes only corresponding first N-bit instructions, i.e., instructions [MOVEQ R2, R6], [MOVEQ R3, R7], and [MOVEQ R4, R8], since the flag 310 is set to "true". Next, the processor continuously executes the general M-bit instruction (6) as there is no more parallel condition execution instruction.

Page 7, line 23 to Page 8, line 8:

When the processor executes the instruction (1) and the content of register R1 is not [[0]] zero, the

comparison obtains different results, so that the result performed on the condition execution instruction is condition rejection. Therefore, the flag 310 is set to "false". At this point, when the processor executes the parallel condition execution instruction (2)~(5), the processor finds the flag as "false" and thus executes only corresponding second N-bit instructions, i.e., [MOVNE R1, R9], [MOVNE R2, R10], [MOVNE R3, R11], and [MOVNE R4, R12]. Next, the processor continuously executes the general M-bit instruction (6) as there is no more parallel condition execution instruction.

<u>Page 8, lines 9-17</u>:

FIG. 6 is a schematic view of another embodiment in accordance with the invention. In FIG. 6, additional instructions can be presented between <u>the</u> condition execution instruction (instruction (1)) and <u>the</u> parallel condition execution instruction (instruction (3)) without affecting the flag. When the processor executes <u>the</u> instruction (1), the flag 310 is set based on the result performed on the instruction (1). Because the instruction (2) does not affect the flag, the processor is still based on the flag 310 to select <u>the</u> first N-bit instructions or second N-bit instructions of the parallel condition execution instructions (3) to (6) to execute.

<u>Page 8, line 18 to Page 9, line 2</u>:

FIG. 7 is a schematic view of another embodiment in accordance with the invention. In FIG. 7, additional instructions can be presented between parallel condition execution instructions without affecting the flag. As shown in FIG. 7, when the processor executes the

instruction (1), the flag 310 is set based on the result performed on the instruction (1). Because the instruction (4) does not affect the flag, the processor is still based on the flag 310 to select the first N-bit instruction (MOVEQ R3, R7) or second N-bit instruction (MOVNE R3, R11) of the parallel condition execution instructions (5) to (6) to execute.

Page 9, lines 3-11:

FIG. 8 is a schematic view of another embodiment in accordance with the invention, which shows that the condition execution instruction is an N-bit (N=16) instruction. As shown, when the processor executes a condition execution instruction (CMP R1, 0) in the instruction (1), the flag 310 is set based on the result performed on the instruction. Because the other instruction in the instruction (1) does not affect the flag, the processor is still based on the flag 310 to select the first N-bit instruction (MOVEQ R1, R5) or second N-bit instruction (MOVNE R1, R9) of the parallel condition execution instructions (2) to (5) to execute.